TBMI26 – Computer Assignment Reports  
Reinforcement Learning

Deadline – March 14 2021

Author/-s:

Martynas Lukosevicius

Alejo Perez Gomez

In order to pass the assignment you will need to answer the following questions and upload the document to LISAM. Please upload the document in PDF format. **You will also need to upload all code in .m-file format**. We will correct the reports continuously so feel free to send them as soon as possible. If you meet the deadline you will have the lab part of the course reported in LADOK together with the exam. If not, you’ll get the lab part reported during the re-exam period.

1. **Define the V- and Q-function given an optimal policy. Use equations and describe what they represent. (See lectures/classes)**

a – actions, s – states, r – reward, t – time step, γ- discount factor

V-function shows how good is to be at a specific state

Q function shows how good is to perform specific action in a specific state

1. **Define a learning rule (equation) for the Q-function and describe how it works. (Theory, see lectures/classes)**

After agent chose and made an action Q-Value is updated by taking some part of previous Q value plus part of reward agent received after performing an action and goodness of the state agent appeared after an action.

1. **Briefly describe your implementation, especially how you hinder the robot from exiting through the borders of a world.**

Agent gets penalized after every movement by -0.1. in some regions its penalized by -0.5, when getting to terminal point it’s not penalized. Agent is not penalized additionally for hitting the wall.

Agent is e-greedy so with some probability it will perform random movements (exploration).

After every movement Q matrix is updated by equation stated in question 2.

1. **Describe World 1. What is the goal of the reinforcement learning in this world? What parameters did you use to solve this world? Plot the policy and the V-function.**

Goal is to find route which gives highest sum Q(s,a). In this case it will be avoiding irritating blob.

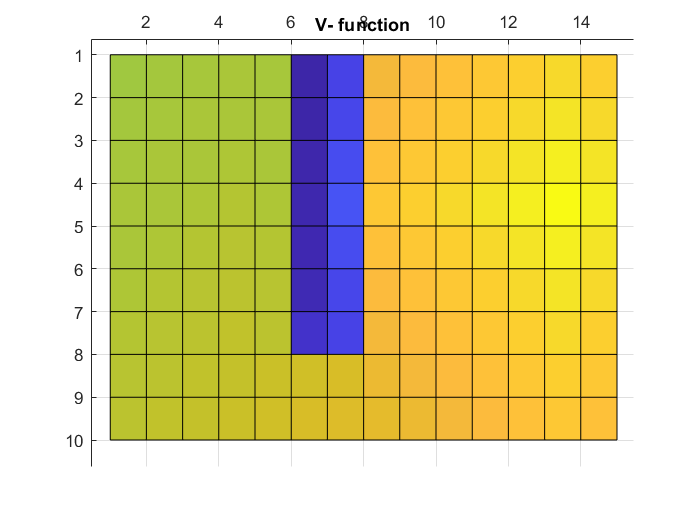
r – as stated in question 3,

maxepochs = 3000;

alpha = 0.6; % learning rate

gamma = 0.9; % discount rate

eps = 0.6; % exploration rate

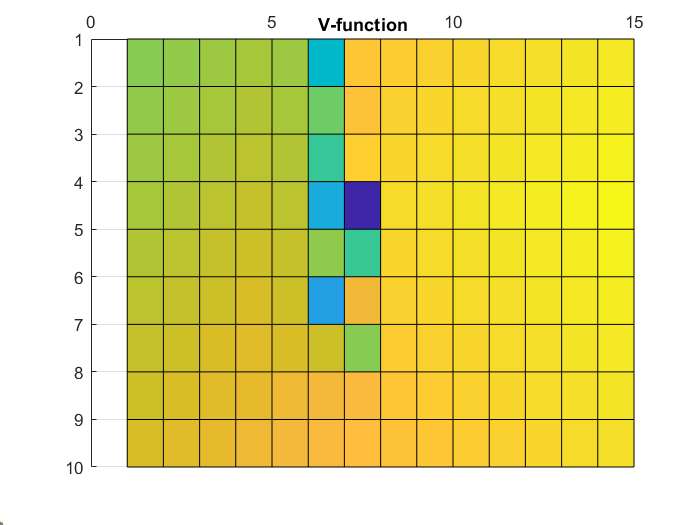
1. **Describe World 2. What is the goal of the reinforcement learning in this world? This world has a hidden trick. Describe the trick and why this can be solved with reinforcement learning. What parameters did you use to solve this world? Plot the policy and the V-function.**

Goal is the same:to find route which gives highest sum Q(s,a). Sometimes some specific states gives bigger penalty than usual, and agent tries to avoid it.

maxepochs = 10000;

alpha = 0.6; % learning rate

gamma = 0.9; % discount rate

**** ****

1. **Describe World 3. What is the goal of the reinforcement learning in this world? Is it possible to get a good policy from every state in this world, and if so how? What parameters did you use to solve this world? Plot the policy and the V-function.**

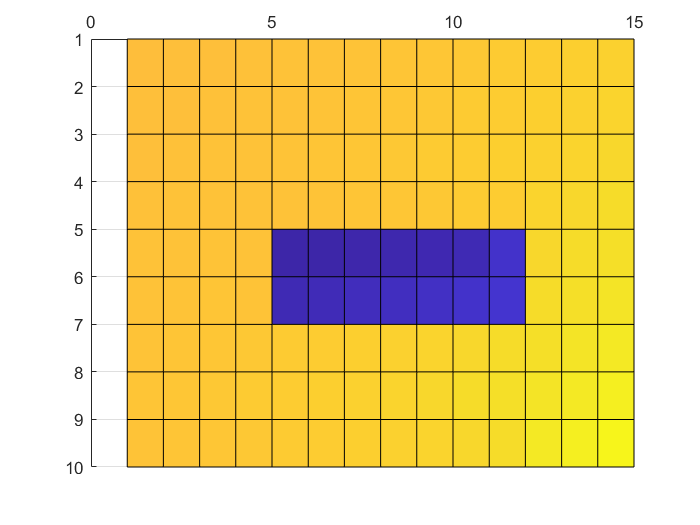
Goal is the same:to find route which gives highest sum Q(s,a). as agent always starts from same position, it is necessary to explore the world. This is done by e-greedy policy, It means that with some probability agent will perform random action.

maxepochs = 10000;

alpha = 0.6; % learning rate

gamma = 0.9; % discount rate

eps = 0.6 % exploration rate

**** ****

1. **Describe World 4. What is the goal of the reinforcement learning in this world? This world has a hidden trick. How is it different from world 3, and why can this be solved using reinforcement learning? What parameters did you use to solve this world? Plot the policy and the V-function.**

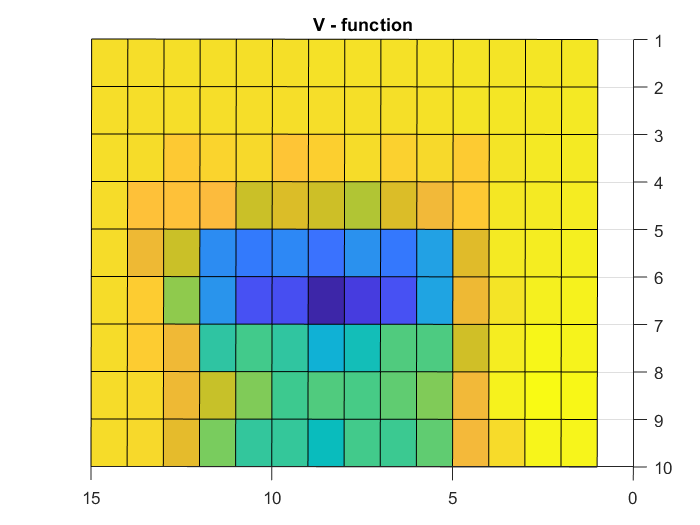
Same as world 3 just agent is pushed to irritating blob, that’s why it learned to go as far as possible from the blob. This can be solved by reinforcement learning because it tries to find optimal route over many trials.

maxepochs = 6000;

alpha = 0.1; % learning rate

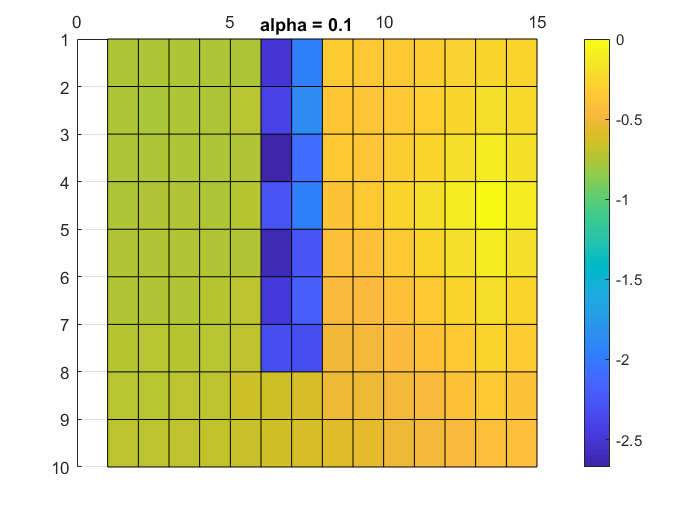
gamma = 0.9; % discount rate

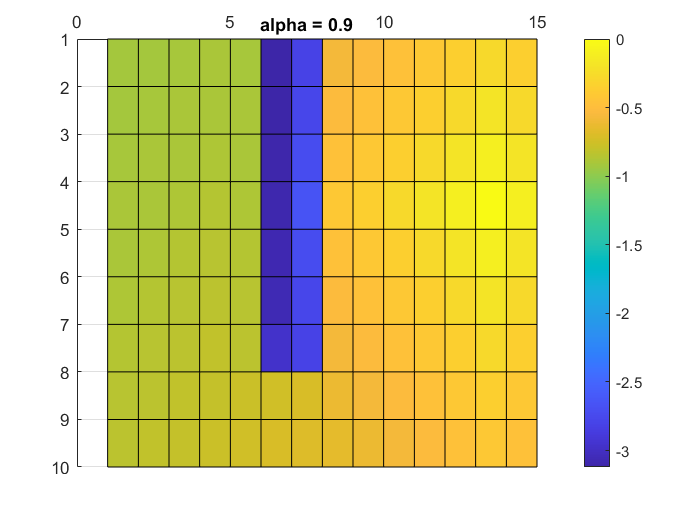
eps = 0.6; % exploration rate

**** ****

1. **Explain how the learning rate α influences the policy and V-function. Use figures to make your point.**

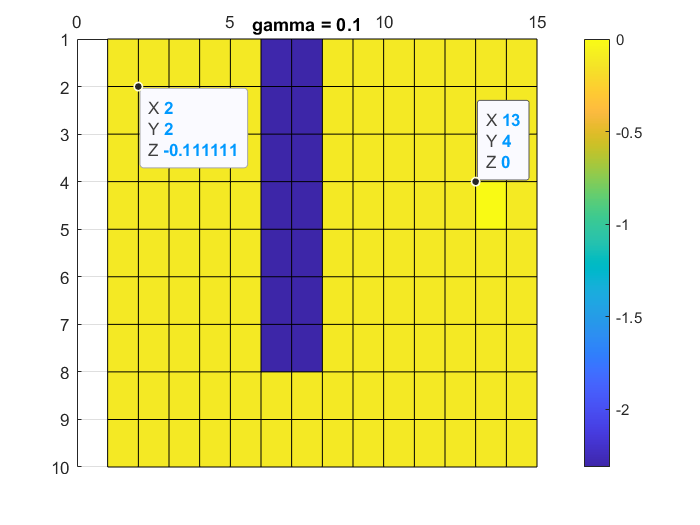
Lower learning rate makes Q values more stable for randomness, however agent will need more time to converge. In the given plots’ agent was run 500 times. Its visible that agent with alpha = 0.9 is closer to convergence than agent with alpha = 0.1.

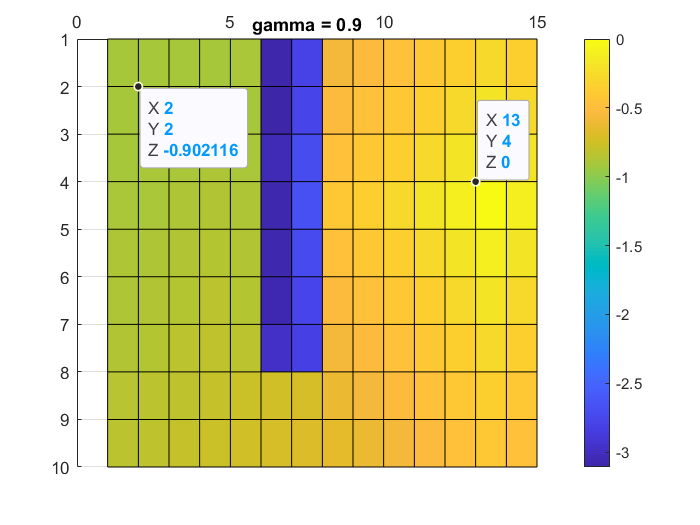
** **

** **

1. **Explain how the discount factor γ influences the policy and V-function. Use figures to make your point.**

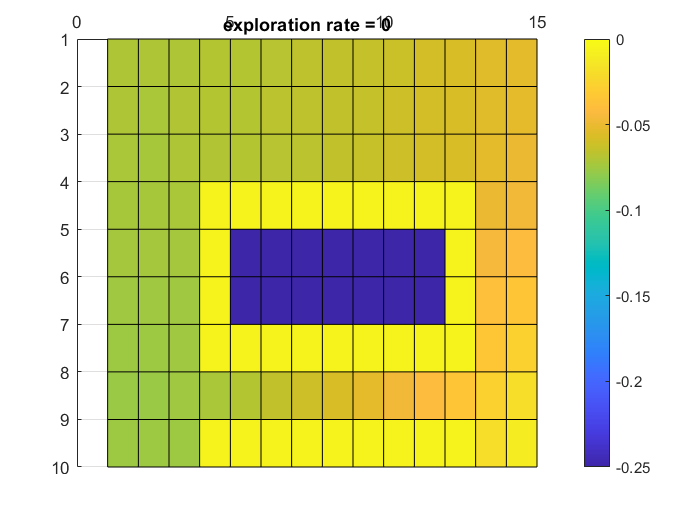
Discount factor shows how eager agent is to reach terminal node. With high factor agent will ignore rewards/penalties and go straight to the state which has highest V – value. As we can see in the V-function plots, there is bigger difference in values of states with higher gamma value.

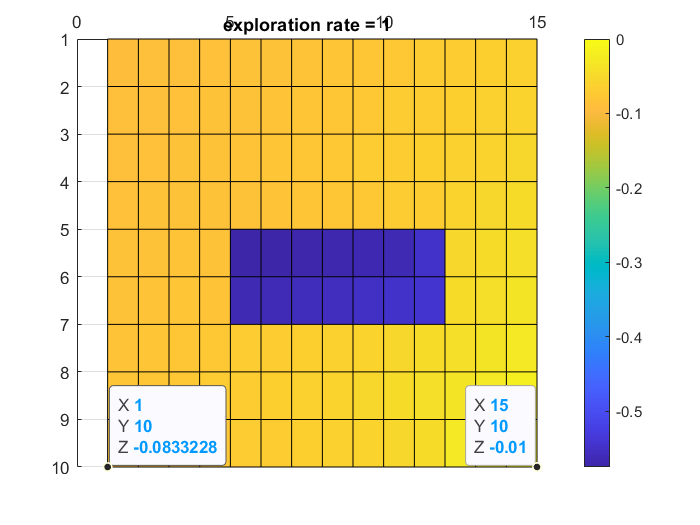
** **

****

1. **Explain how the exploration rate ε influences the policy and V-function. Use figures to make your point. Did you use any strategy for changing ε during training?**

With high exploration rate agent will explore more states and as the result will find best result and will converge faster. Agent with small exploration rate won’t be exploring and will go over the states which were already visited. As we can see in the plots, policy of agent with high exploration rate is converged (first image), while with small exploration rate its opposite (second image).

****

** **

1. **What would happen if we instead of reinforcement learning were to use Dijkstra's cheapest path finding algorithm in the ''Suddenly irritating blob'' world? What about in the static ''Irritating blob'' world?**

Dijkstra's cheapest path finding algorithm consists in trying to find the minimum cost path between the initial point until the terminal node. There has to be an evaluation of all neighbor vertices at each state, whenever the cheapest end-to-end path is obtained, the algorithm stops. The evaluation of Q for all possible states and actions would imply an enormous computational cost compared to the reinforcement learning approach by means of the feedback provided to an agent.

After a possible convergence in the “suddenly irritating blob world”, after evaluating all possible neighboring actions would possibly go through the irritating (more costly) states. This scenario would be originated whether passing through this dangerous area is cheaper than rounding it. This scenario will be more likely to happen in the case of “suddenly irritating blob world” rather than the “irritating blob world” (where the irritating area has more extension).

1. **Can you think of any application where reinforcement learning could be of practical use? A hint is to use the Internet.**

Reinforcement learning can solve optimization problems (tuning parameters etc.). Robotics – learning some behaviors like avoiding obstacles, walking etc. Learn to play computer games.

1. **(Optional) Try your implementation in the other available worlds 5-12. Does it work in all of them, or did you encounter any problems, and in that case how would you solve them?**